Probabilistic Inference in Hybrid Domains by Weighted Model Integration

Vaishak Belle
Dept. of Computer Science
KU Leuven, Belgium
vaishak@cs.kuleuven.be

Andrea Passerini
DISI
University of Trento, Italy
passerini@disi.unitn.it

Guy Van den Broeck
Dept. of Computer Science
KU Leuven, Belgium
guy.vandenbroeck@cs.kuleuven.be

Abstract

Weighted model counting (WMC) on a propositional knowledge base is an effective and general approach to probabilistic inference in a variety of formalisms, including Bayesian and Markov Networks. However, an inherent limitation of WMC is that it only admits the inference of discrete probability distributions. In this paper, we introduce a strict generalization of WMC called weighted model integration that is based on annotating Boolean and arithmetic constraints, and combinations thereof. This methodology is shown to capture discrete, continuous and hybrid Markov networks. We then consider the task of parameter learning for a fragment of the language. An empirical evaluation demonstrates the applicability and promise of the proposal.

1 Introduction

Weighted model counting (WMC) is a basic reasoning task on propositional knowledge bases. It extends the model counting task, or #SAT, which is to count the number of satisfying assignments (that is, models) to a given logical sentence [Gomes *et al.*, 2009]. In WMC, one accords a weight to every model, and computes the sum of the weights of all models. The weight of a model is often factorized into weights of assignments to individual variables.

The WMC formulation has recently emerged as an assembly language for probabilistic reasoning, offering a basic formalism for encoding various inference problems. State-of-the-art reasoning algorithms for Bayesian networks [Chavira and Darwiche, 2008], their relational extensions [Chavira *et al.*, 2006], factor graphs [Choi *et al.*, 2013], probabilistic programs [Fierens *et al.*, 2013], and probabilistic databases [Suciu *et al.*, 2011] reduce their inference problem to a WMC computation. Exact WMC solvers are based on knowledge compilation [Darwiche, 2004; Muise *et al.*, 2012] or exhaustive DPLL search [Sang *et al.*, 2005]. Approximate WMC algorithms use local search [Wei and Selman, 2005] or sampling [Chakraborty *et al.*, 2014]. More recently, the task has also been generalized to first-order knowledge bases [Van den Broeck *et al.*, 2011; Gogate and Domingos, 2011].

The popularity of WMC can be explained as follows. Its formulation elegantly decouples the logical or symbolic representation from the statistical or numeric representation, which is encapsulated in the weight function. When building solvers, this allows us to reason about logical equivalence and reuse SAT solving technology (such as constraint propagation and clause learning). WMC also makes it more natural to reason about deterministic, hard constraints in a probabilistic context. Nevertheless, WMC has a fundamental *limitation*: it is purely Boolean. This means that the advantages mentioned above only apply to *discrete probability distributions*.

A similar observation can be made for the classical satisfiability (SAT) problem and related tasks, which for the longest time could only be applied in discrete domains. This changed with the increasing popularity of *satisfiability modulo theories* (SMT), which enable us to, for example, reason about the satisfiability of linear constraints over the rationals.

This paper generalizes the weighted model counting task to hybrid domains. The resulting *weighted model integration* task (WMI) is defined on the models of an SMT theory Δ, containing mixtures of Boolean and continuous variables. For every assignment to the Boolean and continuous variables, the WMI problem defines a density. The WMI for Δ is computed by integrating these densities over the domain of solutions to Δ, which is a mixed discrete-continuous space. Consider, for example, the special case when Δ has no Boolean variables, and the weight of every model is 1. Then, WMI simplifies to computing the volume of the polytope encoded in Δ [Ma *et al.*, 2009]. When we additionally allow for Boolean variables in Δ, this special case becomes a hybrid version of #SAT [Luu *et al.*, 2014; Chistikov *et al.*, 2015].[1]

To illustrate WMI, we explore its application to inference and learning in *hybrid Markov networks*. Existing inference algorithms for hybrid graphical models are either approximate (*e.g.,* Murphy [1999] or Lunn *et al.* [2000]), or they make strong assumptions on the form of the potentials, such as Gaussian distributions [Lauritzen and Jensen, 2001]. The need for novel approaches to hybrid graphical models is also noted by two approaches that use *piecewise-polynomial* potentials: Shenoy and West [2011] generalize the join-tree algorithm and Sanner and Abbasnejad [2012] generalize symbolic variable elimination. We also consider the piecewise-polynomial setting in this paper, but in a very general frame-

[1] In an independent effort, Chistikov *et al.* [2015] introduce a related definition for the *unweighted model counting* of SMT theories, which they refer to as #SMT. Moreover, their focus is on approximate counting and they do not consider parameter learning.

work where constraints can be defined over arbitrary Boolean connectives. This setting is expressive enough to effectively approximate any continuous distribution, while still permitting exact and efficient computations [De Loera *et al.*, 2012]. In sum, the WMI formulation admits a general and powerful methodology for exact inference in hybrid domains, and is efficient for piecewise-polynomial specifications.

We structure the work as follows. We begin by considering some preliminaries, including the logical language and the standard notion of WMC. We then provide a definition for WMI, consider important classes of specifications and finally turn to parameter learning and empirical evaluations.

2 Preliminaries

We introduce the preliminaries for this work in three steps, beginning with probabilistic models, and then turning to the necessary logical background and WMC.

Probabilistic Models

Let $\mathcal{B}$ and $\mathcal{X}$ denote sets of Boolean and real-valued random variables, that is, $b \in \mathcal{B}$ is assumed to take values from $\{0, 1\}$ and $x \in \mathcal{X}$ takes values from $\mathbb{R}$. We let $(\mathbf{b}, \mathbf{x}) = (b_1, \ldots, b_m, x_1, \ldots, x_n)$ be an element of the probability space $\{0, 1\}^m \times \mathbb{R}^n$, which denotes a particular assignment to the random variables from their respective domains. We let the joint probability density function be denoted by Pr. Then, $\Pr(\mathbf{b}, \mathbf{x})$ determines the probability of the assignment vector.

When these random variables are defined by a set of dependencies, as can be represented using a *graphical model*, the density function can be compactly factorized [Koller and Friedman, 2009]. In this work, we are concerned with undirected graphical models (that is, Markov networks), where the joint density function can be expressed in terms of the cliques of the graph:

$$\Pr(\mathbf{b}, \mathbf{x}) = \frac{1}{Z} \prod_k \phi_k(\mathbf{b}_k, \mathbf{x}_k)$$

where $\mathbf{b}_k$ and $\mathbf{x}_k$ are those random variables participating in kth clique, and $\phi_k(\cdot, \cdot)$ is non-negative, real-valued *potential function*. It is not necessary that ϕ_k denote probabilities, and so Z is a *normalization* constant, also referred to as the *partition function*, defined as:

$$Z \doteq \sum_{b_1} \cdots \sum_{b_m} \int_{x_1} \cdots \int_{x_n} \left[\prod_k \phi_k(\mathbf{b}_k, \mathbf{x}_k) \right] d\mathcal{X}.$$

Logical Background

Propositional satisfiability (SAT) is the problem of deciding whether a logical formula over Boolean variables and logical connectives can be satisfied by some truth value assignment of the Boolean variables. Given a formula ϕ and assignment (or model or world) M, we write $M \models \phi$ to denote *satisfaction.* We write $l \in M$ to denote the literals (that is, propositions or their negations) that are satisfied at M.

A generalization to this decision problem is that of *Satisfiability Modulo Theories* (SMT). In SMT, we are interested in deciding the satisfiability of a (typically quantifier-free) first-order formula with respect to some decidable background theory, such as linear arithmetic over the rationals $\mathcal{LRA}$. Standard first-order models can be used to formulate SMT. For example, $\mathcal{LRA}$ is the fragment of first-order logic over the signature $(0, 1, +, \leq)$ restricting the interpretation of these symbols to standard arithmetic. See Barrett *et al.* [2009] for a comprehensive treatment.

In this paper, the logical language is assumed to be a combination of $\mathcal{LRA}$ and propositional logic, for which satisfaction is defined in an obvious way. We use p, q and r to range over propositional letters, and x, y, z and c to range over constants of the language [Barrett *et al.*, 2009]. So, ground atoms are of the form q, $\neg p$ and $x + 1 \leq y$. For convenience, we also use a ternary version of $\leq$ written $y \leq x \leq z$ to capture intervals. (This is easily seen to not affect the expressive power.)

For our purposes, we also need the notion of *formula abstraction* and *refinement* [Barrett *et al.*, 2009]. Here, a bijection is established between ground atoms and a propositional vocabulary in that propositions are mapped to themselves and ground $\mathcal{LRA}$ atoms are mapped to fresh propositional symbols. Abstraction proceeds by replacing the atoms by propositions, and refinement replaces the propositions with the atoms. We refer to the abstraction of a SMT formula ϕ as ϕ^- and the refinement of a propositional formula ϕ as ϕ^+. Abstraction and refinement are extended to complex formulas in a manner that is homomorphic with respect to the logical operators: $[\phi \vee \psi]^- = \phi^- \vee \psi^-$, $[\neg\phi]^- = \neg[\phi]^-$ and $[true]^- = true$, and likewise for refinement. For example, $[p \vee (x \leq 10)]^-$ is $p \vee q$, and $[p \vee q]^+$ is $p \vee (x \leq 10)$.

Weighted Model Counting

Weighted model counting (WMC) [Chavira and Darwiche, 2008] is a generalization of model counting [Gomes *et al.*, 2009]. In model counting, also known as #SAT, one counts the number of satisfying assignments of a propositional sentence. In WMC, each assignment has an associated weight and the task is to compute the sum of the weights of all satisfying assignments. WMC has applications in probabilistic inference in discrete graphical models.

Definition 1: Given a formula Δ in propositional logic over literals $\mathcal{L}$, and a *weight function* $w : \mathcal{L} \to \mathbb{R}^{\geq 0}$, the *weighted model count* (WMC) is defined as:

$$\text{WMC}(\Delta, w) = \sum_{M \models \Delta} \text{WEIGHT}(M, w)$$

where

$$\text{WEIGHT}(M, w) = \prod_{l \in M} w(l)$$

Intuitively, the weight of a formula is given in terms of the total weight of its models; the weight of a model is defined in terms of the literals that are true in that model.

Example 2: Consider the following Markov network in *feature representation* [Della Pietra *et al.*, 1997; Richardson and Domingos, 2006] over Boolean variables $\{p, q, r\}$:

$$\begin{array}{ll} \neg p \vee \neg q & \\ 0.1 & p \\ 1.2 & p \vee r \\ 2.5 & q \Rightarrow r \end{array}$$

In English: the first feature is a hard constraint that p and q are mutually exclusive, whereas the third feature increases the weight of a world by a factor 1.2 if p or r is true. The weight of a world is 0 if it does not satisfy the hard constraints, and is otherwise equal to the product of the weights of the satisfied features. The sum of the weights of all worlds is the partition function Z. Note that Z equals the WMC of Δ:

$$\Delta = (\neg p \vee \neg q) \wedge (f_1 \Leftrightarrow p \vee r) \wedge (f_2 \Leftrightarrow \neg q \vee r)$$

with weights: $w(p) = 0.1$, $w(f_1) = 1.2$, $w(f_2) = 2.5$ and for all other symbols, the weight is 1. □

We are often interested in computing the probability of a query q given evidence e in a Boolean Markov network N, for which we use:

$$\Pr_N(q \mid e) = \frac{\text{WMC}(q \wedge e \wedge \Delta, w)}{\text{WMC}(e \wedge \Delta, w)} \tag{1}$$

where Δ encodes N and w encodes the potential. The following correctness result is straightforward to establish:

Theorem 3: *Let N be a Markov network over Boolean random variables $\mathcal{B}$ and potentials $\{\phi_1, \ldots, \phi_k\}$. Let Δ and w be the corresponding encodings over Boolean variables $\mathcal{B}$. Then for any $q, e \in \mathcal{B}$, the probability $\Pr_N(q \mid e)$ is given by* (1).

Proof (Sketch): An assignment to the random variables in N can be exactly mapped to a corresponding model of Δ. In that regard, the probability of the assignment equals the weight of the model, and so the partition function equals the sum of weights of all models of Δ, and the case of the conditional probability is a simple corollary. ∎

Finally, we remark that generalizing Boolean random variables to arbitrary discrete ones, that is, where these variables take values from finite sets is also easily achieved [Sang *et al.*, 2005; Chavira and Darwiche, 2008], and so we restrict all discussions to the Boolean setting without loss of generality.

3 Weighted Model Integration

We are interested in exact inference with a mix of discrete and continuous random variables (under certain limitations). However, an inherent limitation of WMC is that it only admits the inference of discrete probability distributions. That is, because it is based on annotating a propositional theory, it encodes a finite sample space, as appropriate for Boolean or discrete Markov networks.

A Definition

In this section, we introduce *weighted model integration* that is a strict generalization of WMC. The main idea here is to annotate a logical theory with rational and Boolean variables, that is, from a combination of $\mathcal{LRA}$ and propositional logic. Nonetheless, a simple semantic formulation is given based on propositional logic, where, as before, propositional assignments are denoted using M.

Definition 4: Suppose Δ is a SMT theory over Boolean and rational variables $\mathcal{B}$ and $\mathcal{X}$, and literals $\mathcal{L}$. Suppose $w : \mathcal{L} \to \text{EXPR}(\mathcal{X})$, where $\text{EXPR}(\mathcal{X})$ are expressions over $\mathcal{X}$. Then the *weighted model integral* (WMI) is defined as:

$$\text{WMI}(\Delta, w) = \sum_{M \models \Delta^-} \text{vol}(M, w)$$

where

$$\text{vol}(M, w) = \int_{\{l^+ : l \in M\}} \text{weight}(M, w)\, d\mathcal{X}$$

The intuition is this. The WMI of a SMT theory Δ is defined in terms of the models of its propositional abstraction Δ^-. For each such model, we compute its volume, that is, we integrate the w-values of the literals that are true at the model. The interval of the integral is defined in terms of the refinement of the literal. The w-function is to be seen as mapping an expression e to its *density function*, which is usually another expression mentioning the variables in e.

Let us remark that while the interval is defined in terms of SMT literals, this is meant to denote standard integrals in an obvious fashion; for example:

$$\int_{x \leq 6} \phi dx \doteq \int_{-\infty}^{6} \phi dx; \qquad \int_{5 \leq x \leq 6} \phi dx \doteq \int_{5}^{6} \phi dx$$

If the subscript is a propositional literal, then it is simply ignored. In general, the standard integral for a $\mathcal{LRA}$-formula δ over variables $x_1, \ldots, x_n$ can be obtained by:

$$\int_{\delta} \phi dx \doteq \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \mathbb{I}_{\delta} \times \phi \; dx_1 \cdots dx_n$$

where $\mathbb{I}_e$ is the indicator function for the event e. So, for example, we obtain the following equivalence:

$$\int_{x \leq 6 \wedge y \geq 5} \phi dx dy = \int_{\mathbb{R}} \int_{\mathbb{R}} \mathbb{I}_{(x \leq 6 \wedge y \geq 5)} \times \phi \; dx dy = \int_{x \leq 6} \int_{y \geq 5} \phi dx dy.$$

Conditional Probabilities

As usual, given a Markov network N over Boolean and real-valued random variables, conditional probabilities are obtained using:

$$\Pr_N(q \mid e) = \frac{\text{WMI}(\Delta \wedge q \wedge e, w)}{\text{WMI}(\Delta \wedge e, w)} \tag{2}$$

where Δ and w denote the network and potential encodings.

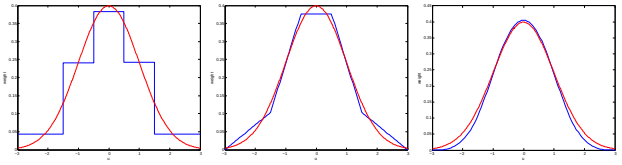
To see WMI in action, consider a simple example:

Example 5: Suppose Δ is the following formula:

$$p \vee (0 \leq x \leq 10)$$

For weights, let $w(p) = .1$, $w(\neg p) = 2x$, $w(q) = 1$ and $w(\neg q) = 0$, where q is the propositional abstraction of $(0 \leq x \leq 10)$. Roughly, this can be seen to say that x is uniformly distributed when p holds and otherwise it is characterized by a triangular distribution in the interval [0, 10]. There are 3 models of Δ^-:

1. $M = \{p, \neg q\}$: since $w(\neg q) = 0$, by definition we have $\text{weight}(M, w) = 0$ and so $\text{vol}(M, w) = 0$;
2. $M = \{\neg p, q\}$: $\text{vol}(M, w) = \int_{0 \leq x \leq 10} 2x \, dx = \left[x^2\right]_0^{10} = 100$.
3. $M = \{p, q\}$: $\text{vol}(M, w) = \int_{0 \leq x \leq 10} .1 \, dx = [.1 \cdot x]_0^{10} = 1$.



(a) constants (b) degree 1 (c) degree 3

Figure 1: Approximations (blue) to Gaussians (red).

Thus, $\text{WMI}(\Delta, w) = 100 + 1 = 101$.

Suppose that we are interested in the probability of the query $x \leq 3$ given that $\neg p$ is observed. Suppose r is the abstraction of $x \leq 3$. First, $\text{WMI}(\Delta \wedge \neg p, w)$ corresponds to the weight of a single interpretation, that of item 2, yielding a value of 100. Next, $\text{WMI}(\Delta \wedge \neg p \wedge x \leq 3, w)$ also corresponds to the weight of a single interpretation $M = \{\neg p, q, r\}$, an extension to that in item 2. In this case:

$$\text{VOL}(M, w) = \int_{(0 \leq x \leq 10) \wedge (x \leq 3)} 2x \, dx = \left[x^2\right]_0^3 = 9.$$

Therefore, the conditional probability is $9/100 = .09$. □

Generality

Recall that propositions as intervals to integrals are simply ignored, and propositional logic is a fragment of the logical language considered here. So, we get:

Proposition 6: *Suppose Δ is a formula in propositional logic over literals $\mathcal{L}$ and $w : \mathcal{L} \to \mathbb{R}^{\geq 0}$. Then* $\text{WMI}(\Delta, w) = \text{WMC}(\Delta, w)$.

Classes of Weight Functions

The weight function in the WMI formulation does not commit to any restrictions on $\text{EXPR}(\mathcal{X})$, in which case arbitrary continuous functions (including exponential families) can be encoded. In practice, it will be useful to identify two special cases. We are motivated by inference for two classes of representations: *piecewise-constant hybrid Markov networks* (CHMN) and *piecewise-polynomial hybrid Markov networks* (PHMN). These are Markov networks in feature representation, as shown before, where now the features are SMT sentences, and the weights are constants and polynomials respectively. That is, inference for CHMNs can be reduced to WMI where $w : \mathcal{L} \to \mathbb{R}^{\geq 0}$, and for PHMNs, w maps $\mathcal{L}$ to polynomials over $\mathcal{X}$.

Although CHMN subsume discrete Markov networks, they are still very limited in the continuous distributions they can represent. In Example 5, a uniform distribution was represented, but consider the case of a Gaussian:

Example 7: Suppose u is normally distributed with mean 0 and variance 1, conditioned to be in the three-deviation interval $[-3, 3]$. We can approximate[2] this distribution with the following CHMN:

$$\begin{array}{ll} & -3 \leq u \leq 3 \\ 0.043 & u \leq -1.5 \\ 0.241 & -1.5 < u \leq -0.5 \\ 0.383 & -0.5 < u \leq 0.5 \\ 0.241 & 0.5 < u \leq 1.5 \\ 0.043 & 1.5 < u \end{array}$$

whose density is depicted in Figure 1a. The unweighted formula is a hard constraint. □

PHMNs, of course, give us considerably more expressive power in the weight function:

Example 8: Suppose u is as before, but approximated using the following PHMN:

$$\begin{array}{rl} & -3 \leq u \leq 3 \\ (2+u)^3/6 & -2 < u \leq -1 \\ (4 - 6u^2 - 3u^3)/6 & -1 < u \leq 0 \\ (4 - 6u^2 + 3u^3)/6 & 0 < u \leq 1 \\ (2-u)^3/6 & 1 < u < 2 \end{array}$$

which appeals to polynomials of degree 3, and is plotted in Figure 1c. We might compare it to Figure 1b that shows a coarser approximation than this one using polynomial weights of degree 1. □

Correctness

Finally, we provide an appropriate companion to Theorem 3:

Theorem 9: *Let N be a Markov network over Boolean and real-valued random variables $\mathcal{B}$ and $\mathcal{X}$ and potentials $\{\phi_1, \ldots, \phi_k\}$. Let Δ and w be the corresponding encodings. Then for any $q, e \in \mathcal{B} \cup \mathcal{X}$, $\Pr_N(q \mid e)$ is given by* (2).

Proof (Sketch): The argument is similar in spirit to the one provided for Theorem 3. For any N, suppose ρ is the density accorded to all assignments $(\mathbf{b}, \mathbf{x})$ of the random variables in $\mathcal{B} \cup \mathcal{X}$, where $c_1 \leq x_1 \leq d_1, \ldots, c_n \leq x_n \leq d_n$. Then the subset of the probability space obtained from $b_1 \times \cdots \times b_n \times [c_1, d_1] \times \cdots \times [c_n, d_n]$ can be exactly mapped to a model of Δ^-, in the sense of being accorded the same density. (The assumption here, of course, is that Δ would encode these intervals as SMT literals of the form $c_i \leq x_i \leq d_i$ with an appropriate w-value.) The partition function Z is calculated by integrating over $\mathbb{R}^n$ for all such subsets. Using the definition:

$$\int_{\{l^+ : l \in M\}} \text{WEIGHT}(M, w) d\mathcal{X} = \int_{\mathbb{R}^n} \mathbb{I}_{\{l^+ : l \in M\}} \times \text{WEIGHT}(M, w) \, d\mathcal{X}$$

together with the observation:

$$\text{WMI}(\Delta, w) = \int_{\mathbb{R}^n} \sum_{M \models \Delta^-} \mathbb{I}_{\{l^+ : l \in M\}} \times \text{WEIGHT}(M, w) \, d\mathcal{X}$$

we find that $\text{WMI}(\Delta, w)$ is also obtained by integrating over $\mathbb{R}^n$ for all corresponding models of Δ^-. Thus, $\text{WMI}(\Delta, w) = Z$, and the case of the conditional probability is a simple corollary. ∎

[2] The weights were computed using the software MATLAB (`www.mathworks.com`) to be the least-squares approximation.

Inference Complexity

To conclude the section, let us discuss the complexity of the volume computations that need to be performed for WMI. One may be concerned about the degree of the polynomials being integrated. Indeed, these degrees can become large.

Proposition 10: *Suppose Δ, $\mathcal{B}$, $\mathcal{X}$, $\mathcal{L}$ and w is as before, where w encodes a* PHMN. *Let k be the maximum degree of the polynomials in $\{w(l) \mid l \in \mathcal{L}\}$. Let n be the number of propositions in Δ^{-}. Then* $\text{WMI}(\Delta, w)$ *computes volumes of polynomials of degree $k \cdot n$.*

Proof: Suppose M is a model of Δ^{-}. By assumption $size(\{l^{+} \mid l \in M\}) = n$. Then $\text{WEIGHT}(M, w)$ would be a product of n polynomials each of degree k, giving us $(k \cdot n)$. ∎

However, Baldoni *et al.* [2011] show that for a fixed number of variables, the integration is efficient, even for polynomials of high degree.[3] Smoother function approximations therefore come at a reasonable cost.

4 Parameter Learning for CHMN

This section focuses on the weight learning task for CHMNs. Weight learning uses data $\mathcal{D}$ to automatically learn the density associated with each feature (that is, an SMT formula) by optimizing a given objective function. This section explores how the standard practice of parameter learning [Koller and Friedman, 2009] can be understood for our framework.[4]

Let us recall some preliminaries. We appeal to the notion of *maximum likelihood estimation* where given a set of parameters $w_1, \ldots, w_k$, and $\mathcal{D}$, we seek to maximize the likelihood (MLE) of these parameters given the data:

$$\max_{\mathbf{w}} L(\mathbf{w} : \mathcal{D})$$

For technical reasons, given a Markov network N, a *log-linear model* for the joint density function is often considered, as follows:

$$\Pr_N(\mathbf{b}, \mathbf{x}) = \frac{1}{Z} \exp\left(\sum_k w_k \cdot f_k(\mathbf{b}_k, \mathbf{x}_k)\right),$$

where clique potentials are replaced by an exponentiated weighted sum of *features*. Then, the likelihood can be shown to be maximum precisely when:

$$\mathrm{E}_{\mathcal{D}}[f_i] = \mathrm{E}_{\mathbf{w}}[f_i]$$

which says the empirical expectation of f_i in $\mathcal{D}$ (for example, the count) equals its expectation according to the current parameterization.

In our setting, we first observe that for PHMNs, taking a logistic model would mean that the density function is the Euler number e raised to a complex high degree polynomial,[5] which would be prohibitive in practice.[6] We focus instead on CHMNs, which can be given a fully logistic formulation.

Given $\mathcal{D}$ and a SMT theory Δ, the empirical expectation of a formula $\alpha \in \Delta$ can be obtained by calculating:

$$\mathrm{E}_{\mathcal{D}}[\alpha] = size(\{d \mid d \models \alpha \text{ and } d \in \mathcal{D}\}) \,/\, size(\mathcal{D})$$

That is, we simply count the items in $\mathcal{D}$ which are true for α. For example, if $(x = 3.5) \in \mathcal{D}$ and $(p \vee x \leq 7) \in \Delta$, then the count of the feature $(p \vee x \leq 7)$ increases.

Given a SMT theory Δ and a weight function w, the expectation of $\alpha \in \Delta$ wrt w is given by

$$\mathrm{E}_w[\alpha] = \text{WMI}(\alpha, w) \,/\, Z$$

With this formulation, standard iterative methods from convex optimization can be used [Koller and Friedman, 2009]. Building on Theorem 9, we prove:

Theorem 11: *Suppose N is a Markov network provided as a log-linear model (with numeric weights). Given its encoding as an SMT theory Δ, the parameter estimates for $\alpha \in \Delta$ maximizing the likelihood given $\mathcal{D}$ are those minimizing the difference between* $\mathrm{E}_{\mathcal{D}}[\alpha]$ *and* $\mathrm{E}_w[\alpha]$.

5 Experimental Evaluations

In this section, we discuss results on an implementation of WMI inference and CHMN parameter learning. The WMI implementation scheme is an exhaustive DPLL search modulo an SMT oracle.[7]

Our system is implemented using the Z3 SMT solver v4.3.2,[8] and the LattE software v1.6 for computing integrals.[9] All experiments were run using a system with 1.7 GHz Intel Core i7 and 8GB RAM.

Scaling Behavior

To test the scaling behavior, we are mainly concerned with the volume computation aspect of the WMI task. Observe that WMI for Δ can be seen to implicitly compute #SAT on Δ^{-}. Therefore, to evaluate our implementation, the system is also made to compute $\#\text{SAT}(\Delta^{-})$, which can be enabled by simply letting $\text{VOL}(M, w) = 1$ for any M and w.

For our tests, we randomly generated SMT theories and weight functions, involving intricate dependencies and hard constraints. These included weighted sentences of the form:

$$\begin{aligned} x^3 \quad & f_1 \Leftrightarrow [x + 3y \leq 3] \\ .001y^2 - .11y + 2.5 \quad & f_2 \Leftrightarrow [p \vee (x \geq 0 \wedge y \geq 0)] \\ .1 \quad & f_3 \Leftrightarrow (\neg p \vee \neg q) \end{aligned}$$

with additional hard constraints of the sort:

$$(f_1 \wedge f_2) \Rightarrow \neg f_3.$$

[3] See Baldoni *et al.* [2011] also for discussions on how this relates to the more general problem of computing volumes of arbitrary polytopes [Dyer and Frieze, 1988].

[4] While we consider maximum likelihood estimation here, max-margin learning of weighted SMT theories has also been recently proposed [Teso *et al.*, 2015].

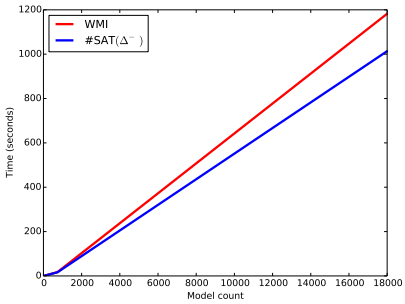
[5] No approximations for Gaussians would then be needed.

[6] The MLE formulation can be given for non-logistic models under certain reasonable assumptions, which can be applied to PHMNs. We leave this to an extended version of the paper.

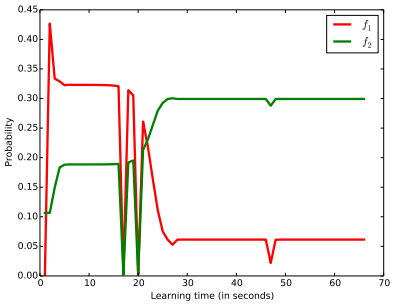
[7] We remark that our implementation does not yet consider the full range of effective WMC techniques [Sang *et al.*, 2005], such as component caching [Bacchus *et al.*, 2009]. This is an interesting avenue for the future.

[8] `http://z3.codeplex.com`

[9] `https://www.math.ucdavis.edu/~latte`



(a) scaling behavior



(b) parameter learning

Query	Probability
$\Pr(J < 1000 \mid M)$	0.449245
$\Pr(J < 1000 \mid M \wedge F)$	0.063354
$\Pr(J < 1000 \mid M \wedge F \wedge G)$	0.226589

where:

$J = j_1 + j_2 + j_3$;
$M = morn \wedge j_1 > 300$;
$F = j_1 > 320 \wedge fri$;
$G = (s_3 >= 95)$

(c) conditional queries

Figure 2: Empirical evaluations.

While iteratively increasing the number of variables and constraints in the SMT theories, we plot the WMI behavior and use #SAT(Δ^-) as the baseline against all instances of a certain model count in Figure 2a. (The plots are smoothed for readability.) This compares, for example, the effort for weight and volume computations as the number of models of the input theory increases. We see that the implementation scales well relative to #SAT(Δ^-). While our theories sparingly mention high-degree polynomials, integration is still necessary for almost all real-valued variables and so this demonstrates that WMI and its volume computation approach is both feasible and can be made effective.

Real-World Dataset

Next, we demonstrate parameter learning, and the diversity of applications that WMI can characterize, well beyond the standard hybrid examples. We consider the following novel application involving conditional queries over arithmetic constraints. It uses a data series released by the UK government that provides average journey time, speed and traffic flow information on all motorways, known as the Strategic Road Network, in England.[10] Motorways are split into junctions, and each information record refers to a specific junction, day and time period. In the following we consider the 2012 dataset, with over 7 million entries, and focus on a set of junctions surrounding Heathrow airport (J12-J16).

The task is as follows. Think of a commuter going to work. She expresses assumptions about her journey in terms of the time period of the journey, its duration, the average speed that she can drive at through a junction using soft constraints:

$$f_1 \Leftrightarrow \quad [morn \Rightarrow j_1 + j_2 + j_3 \leq 800]$$
$$f_2 \Leftrightarrow \quad [aft \Rightarrow avg(s_1, s_2, s_3) \geq 80]$$
$$f_3 \Leftrightarrow \quad [morn \vee eve \Rightarrow 700 \leq (j_1 + j_2 + j_3) \leq 900]$$

which can be read as if her travel is during the morning, the total journey time is (hopefully) less than 800 seconds, that the average speed exceeds 80 kmh in the afternoon, and so on. Together with hard constraints, such as:

$$0 \leq j_1 + j_2 + j_3 \leq 1500$$

she is interested in complex conditional queries about the journey time given certain conditions such as the day of week and the time of travel.

In our experiments, we applied the parameter learning formulation to learn the weights of such assumptions against the 2012 dataset, mapping terms in the theory to actual data points as appropriate. The weights are initialized to 1, and Figure 2b plots how the weights diverge for the formulas f_1 and f_2 when likelihood estimation terminates. In essence, mornings being peak traffic hours, it is unlikely that a morning journey lasts less than 800 seconds, and so f_1 is accorded a low probability. In contrast, the commuter's afternoon assumptions are reasonable relative to the data, leading to f_2 have a high(er) probability.

Finally, Figure 2c shows how the conditional probability of getting to work on time changes according to the increasing amount of evidence. We observe that the query can be conditioned wrt arbitrary Boolean combinations of arithmetic and propositional constraints. Consider, for example, the second query. After taking into account that at least 320 seconds has been currently spent in the first junction on a Friday morning (*i.e.,* evidence $M \wedge F$), the driver concludes that the probability of getting to work on time ($J < 1000$) is now fairly low. She then refers to a colleague who has passed through the last junction to report that the current average speed in the junction j_3 is over 95 kmh. With this evidence, the probability of getting to work on time increases.

6 Conclusions

WMC is a prominent and general approach to probabilistic inference in Boolean (and discrete) graphical models. In the recent years, the performance of SAT solvers has greatly improved, not only due to massive engineering efforts, but also because of theoretical insights on SAT algorithms themselves. While WMC benefits from these developments, it remains restricted to discrete probability distributions. In this work, we proposed the generalization WMI, proved its correctness, expressiveness as well as its downward compatibility with WMC. It is based on SMT technology that has also enjoyed tremendous progress together with SAT. The notion allows us to address exact inference in mixtures of discrete

[10] http://data.gov.uk/dataset/dft-eng-srn-routes-journey-times

and continuous probability distributions. We then demonstrated parameter learning on a real-world dataset. The results are promising and the formulation is general. For the future, we would like to generalize algorithmic insights from WMC, such as component caching [Bacchus *et al.*, 2009], for the WMI task.

Acknowledgements

Vaishak Belle is partially funded by the Research Foundation-Flanders (FWO-Vlaanderen) project on Data Cleaning and the KU Leuven's GOA on Declarative Modeling for Mining and Learning. Guy Van den Broeck is supported by the Research Foundation-Flanders (FWO-Vlaanderen).

References

[Bacchus *et al.*, 2009] F. Bacchus, S. Dalmao, and T. Pitassi. Solving #SAT and Bayesian inference with backtracking search. *JAIR*, 34(1):391–442, 2009.

[Baldoni *et al.*, 2011] V. Baldoni, N. Berline, J. De Loera, M. Köppe, and M. Vergne. How to integrate a polynomial over a simplex. *Mathematics of Computation*, 80(273):297–325, 2011.

[Barrett *et al.*, 2009] C. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability*. IOS Press, 2009.

[Chakraborty *et al.*, 2014] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi. Distribution-aware sampling and weighted model counting for sat. In *Proc. AAAI*, 2014.

[Chavira and Darwiche, 2008] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, April 2008.

[Chavira *et al.*, 2006] M. Chavira, A. Darwiche, and M. Jaeger. Compiling relational Bayesian networks for exact inference. *Int. Journal of Approximate Reasoning*, 42(1-2):4–20, 2006.

[Chistikov *et al.*, 2015] D. Chistikov, R. Dimitrova, and R. Majumdar. Approximate counting in smt and value estimation for probabilistic programs. In *TACAS*, pages 320–334, 2015.

[Choi *et al.*, 2013] A. Choi, D. Kisa, and A. Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 121–132. Springer, 2013.

[Darwiche, 2004] A. Darwiche. New advances in compiling CNF to decomposable negation normal form. In *Proc. ECAI*, 2004.

[De Loera *et al.*, 2012] J. De Loera, B. Dutra, M. Koeppe, S. Moreinis, G. Pinto, and J. Wu. Software for exact integration of polynomials over polyhedra. *ACM Communications in Computer Algebra*, 45(3/4):169–172, 2012.

[Della Pietra *et al.*, 1997] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[Dyer and Frieze, 1988] M. E. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM J. Comput.*, 17(5):967–974, 1988.

[Fierens *et al.*, 2013] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas. *TPLP*, 2013.

[Gogate and Domingos, 2011] V. Gogate and P. Domingos. Probabilistic theorem proving. In *Proc. UAI*, pages 256–265, 2011.

[Gomes *et al.*, 2009] C. P. Gomes, A. Sabharwal, and B. Selman. Model counting. In *Handbook of Satisfiability*. IOS Press, 2009.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[Lauritzen and Jensen, 2001] S. L Lauritzen and F. Jensen. Stable local computation with conditional gaussian distributions. *Statistics and Computing*, 11(2):191–203, 2001.

[Lunn *et al.*, 2000] D. J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter. Winbugs - a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, 10(4):325–337, 2000.

[Luu *et al.*, 2014] L. Luu, S. Shinde, P. Saxena, and B. Demsky. A model counter for constraints over unbounded strings. In *Proc. ACM SIGPLAN PLDI*, 2014.

[Ma *et al.*, 2009] F. Ma, S. Liu, and J. Zhang. Volume computation for boolean combination of linear arithmetic constraints. In *CADE*, 2009.

[Muise *et al.*, 2012] C. Muise, S. A McIlraith, J. Christopher Beck, and E. I Hsu. Dsharp: fast d-dnnf compilation with sharpsat. In *Advances in Artificial Intelligence*, pages 356–361, 2012.

[Murphy, 1999] K. P Murphy. A variational approximation for bayesian networks with discrete and continuous latent variables. In *Proc. UAI*, pages 457–466, 1999.

[Richardson and Domingos, 2006] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1):107–136, 2006.

[Sang *et al.*, 2005] T. Sang, P. Beame, and H. Kautz. Performing bayesian inference by weighted model counting. In *Proc. AAAI*, 2005.

[Sanner and Abbasnejad, 2012] S. Sanner and E. Abbasnejad. Symbolic variable elimination for discrete and continuous graphical models. In *AAAI*, 2012.

[Shenoy and West, 2011] P. P Shenoy and J. C West. Inference in hybrid bayesian networks using mixtures of polynomials. *Int. Journal of Approximate Reasoning*, 52(5):641–657, 2011.

[Suciu *et al.*, 2011] D. Suciu, D. Olteanu, C. Ré, and C. Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.

[Teso *et al.*, 2015] S. Teso, R. Sebastiani, and A. Passerini. Structured learning modulo theories. *Artif. Intell.*, 2015.

[Van den Broeck *et al.*, 2011] G. Van den Broeck, N. Taghipour, Wannes Meert, J. Davis, and L. De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In *Proc. IJCAI*, pages 2178–2185, 2011.

[Wei and Selman, 2005] W. Wei and B. Selman. A new approach to model counting. In *Theory and Applications of Satisfiability Testing*, pages 96–97. Springer, 2005.